

Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Sähköisen liiketoiminnan järjestelmät

2014

Aleksi Läntinen

TIETOKANTAPOHJAISEN RAPORTINHALLINTAJÄRJES- TELMÄN MÄÄRITTELY JA SUUNNITTELU



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Aleksi Läntinen

TIETOKANTAPOHJAISEN RAPORTINHALLINTAJÄRJESTELMÄN MÄÄRITTELY JA SUUNNITTELU

Tässä opinnäytetyössä määriteltiin ja suunniteltiin raportinhallintajärjestelmä paikallisen yrityksen käyttöön. Yritys tarjoaa kattojen ja muiden rakenteiden kuntoarviointia, joista arvioista tuotetaan asiakkaille tarkat raportit. Asiakkaan aiemmin käyttämällä menetelmillä näiden raporttien valmisteluun on kulunut liikaa aikaa ja niiden laadullinen taso on ollut vaihtelevaa.

Opinnäytetyön kirjallinen osio painottuu ohjelmiston määrittelyyn ja suunnitteluun. Suunniteltu työ kuitenkin myös toteutettiin asiakkaalle. Projektissa käytetiin sekä poisheitettävää prototyyppiä, että evolutiivisia prototyyppejä. Työ tarkastelee valittujen menetelmien ja työkalujen sopivuutta lopulliseen tulokseen ja vaihtoehtoisia toimintamalleja, joilla oltaisiin voitu vähentää tai välttää projektissa ilmenneitä hankaluuksia.

Opinnäytetyössä käytettiin niin määrittely-, suunnittelu-, kuin toteutusvaiheissakin erilaisia Microsoftin Office-pakettiin kuuluvia tuotteita, minkä lisäksi ohjelmistovaatimuksia määriteltäessä prototyökaluna Microsoftin Visual Studiota. Suunnittelua ja toteutusta tukemassa käytettiin Microsoftin omaa ohjekirjastoa, alan kirjallisuutta ja Internetin eri lähteitä keskustelupalstoista koodinäytteisiin. Kehitysmenetelmiin ja -käytäntöihin keskittyneessä teoriaosuudessa on tukena käytetty lisäksi alan kirjallisuutta ja verkkolähteitä.

Työn tuotoksen pohjalta järjestelmä myös toteutettiin asiakkaalle.

ASIASANAT:

Microsoft Access, vaatimusten käsittely, protoilumenetelmä ohjelmistokehityksessä

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme in Business Information Technology | e-Business Systems

15 June 2014 | 36 pages + 3 attachments

Päivi Killström

Aleksi Läntinen

DEFINING AND DESIGNING DATABASE-BASED TOOL FOR HANDLING REPORTS

The purpose of this thesis was to define and design a tool for an easier production and handling of reports for a client. The client offers condition assessments of roofs and other structures. Reports are then produced based on these assessments. The methods used in producing the reports have varied but have in general taken too much working time to prepare and have been of varying quality.

This study emphasizes the requirements of engineering for the tool and the different steps in designing it. The designed tool was also implemented. One of the main objectives of the thesis was to study the theory behind the methods in software development that were applied in the project. In addition, the thesis describes the implementation of the tool in detail whilst also taking note of several possible different approaches and ideas that could have been made use of to reduce the amount of difficulties encountered.

Different products of the Microsoft Office package were used throughout the different phases of the project. Additionally, Microsoft's Visual Studio was used for prototyping the tool in the requirements engineering phase. Source materials used include Microsoft's help libraries, books dealing in software prototyping and requirements engineering and various internet resources.

The designed tool was also implemented and delivered to the client.

KEYWORDS:

Microsoft Access, requirements engineering, software prototyping

SISÄLTÖ

SANASTO	6
1 JOHDANTO	7
2 OHJELMISTOTUOTANNON KÄYTÄNTÖJÄ	8
2.1 Vaatimusten käsittely (requirements engineering)	8
2.1.1 Toiminnalliset vaatimukset	9
2.1.2 Ei-toiminnalliset vaatimukset	10
2.2 Protoilumalli (software prototyping)	10
2.2.1 Poisheitettävät prototyypit (throwaway prototypes)	11
2.2.2 Evoluutioprototyypit (evolutionary prototypes)	12
2.2.3 Protoilumallin haasteet	12
3 CASE: TIETOKANTAPOHJAISEN RAPORTINHALLINTAJÄRJESTELMÄN MÄÄRITTELY JA SUUNNITTELU ASIAKKAALLE	14
3.1 Tarve uudelle raportinhallintajärjestelmälle	14
3.2 Raportinhallintajärjestelmän määrittely	15
3.2.1 Alkupalaveri	15
3.2.2 Käyttöliittymän prototyyppi	15
3.2.3 Raportinhallintajärjestelmän vaatimukset	16
3.2.4 Käyttötapaukset (use cases)	18
3.3 Raportinhallintajärjestelmän suunnittelu	21
3.3.1 Tietokanta	22
3.3.2 Käyttöliittymä- ja logiikka	24
3.3.3 Raportti	29
4 YHTEENVETO JA POHDINTA	32
4.1 Projektin haasteet	32
4.2 Työn lopputulos ja jatkokehityksen tarve	34
4.3 Yhteenveto	35
LÄHTEET	36

LIITTEET

- Liite 1. Prototyypin esittely asiakkaalle
- Liite 2. Tietokanta
- Liite 3. Malliraportti

KUVAT

Kuva 1. Tietokannan rakenne.	23
Kuva 2. Accessin työkalut käyttöliittymän rakentamiseen.	25
Kuva 3. Käyttöliittymäkuva versiosta 0.1.0.	26
Kuva 4. Käyttöliittymäkuva versiosta 0.6.0.	27
Kuva 5. Käyttöliittymäkuva versiosta 0.9.0.	28
Kuva 6. Accessin työkalut raportin rakentamiseen.	29
Kuva 7. Tarkastusraportin muotoilu versioissa 0.1.0 - 0.6.0.	30
Kuva 8. Kustannusarvio versiossa 0.6.0.	31

TAULUKOT

Taulukko 1. Uuden raportin lisääminen.	18
Taulukko 2. Raportin hakeminen ohjelmistosta.	19
Taulukko 3. Toimenpiteiden kustannusarvioiden lisääminen.	20

SANASTO

Access	Access on Microsoftin Office-pakettiin sisältyvä tietokantojen ja tietokantaohjelmistojen suunnittelu- ja toteutustyökalu.
PDF	Portable Document Format eli PDF on Adoben kehittämä ohjelmistoriippumaton, siirrettävä tiedostomuoto, jota käytetään pääasiallisesti sähköiseen julkaisemiseen, tulostamiseen ja painamiseen.
Protoilumenetelmä	Ohjelmistokehityksen menetelmä. Prototyypissä on toteutettu joitakin järjestelmän piirteitä ja sitä voidaan hyödyntää teknisten ratkaisujen kokeiluun tai vaatimusmäärittelyn aputyökaluna. Prototyypistä voidaan myös kehittää lopullinen tuote.
Relaatiotietokanta	Relaatiomallin mukainen tietokanta muodostuu tauluista, jotka sisältävät rivejä ja sarakkeita. Tietokannan tieto perustuu taulujen välille määritellyille yhteyksille. Yhdistävinä tietueina mallissa käytetään erilaisia perus- ja viiteavaimia.
SQL	Structured Query Language eli SQL on standardoitu kyselykieli, jolla voidaan tehdä erilaisia hakuja, muutoksia ja lisäyksiä relaatiotietokantoihin.
UML	Unified Modeling Language eli UML on standardoitu, graafinen mallinnuskieli, jonka käyttämillä eri kaavioilla kuvataan ohjelmiston rakennetta, käyttäytymistä ja eri sidosryhmien vuorovaikutusta.
VBA	Visual Basic for Applications eli VBA on Microsoftin eri sovellusohjelmissa käytetty ohjelmointikieli.
Visual Studio	Visual Studio on Microsoftin ohjelmistonkehitysympäristö, jossa voi käyttää useita eri ohjelmointikieliä. Kehitysympäristö mahdollistaa nopean ja helpon graafisten käyttöliittymien suunnittelun.

1 JOHDANTO

Tämän opinnäytetyön tavoitteena oli määritellä, suunnitella ja toteuttaa raportinhallintajärjestelmä paikalliselle yritykselle. Yrityksen palveluihin kuuluu kattojen ja muiden rakenteiden kuntoarviointi, joista arvioista yritys toimittaa asiakkaalle kattavan raportin. Näiden raporttien valmistelu asiakkaalle tulostamista/lähetttämistä varten on aiemmin toteutettu sekalaisin työvälinein ja -tavoin, mistä syystä eri raporttien ulkoasut ovat poikenneet toisistaan huomattavasti ja vieneet liian paljon työtunteja.

Tehtävänäni oli olla mukana määrittelemässä uutta raportinhallintajärjestelmää aiemmin käytössä olleiden ja tähän tarkoitukseen kankeiden tekstinkäsittely- ja taulukkolaskentaohjelmien korvaajaksi, sekä suunnitella ja toteuttaa tämän vaatimusmäärittelyn mukainen ohjelmisto. Opinnäytetyönä toteutettavan järjestelmän avulla raportin kirjoitus- ja asemointivaiheita oli tarkoitus päästä nopeuttamaan huomattavasti, minkä lisäksi uusi järjestelmä takaisi kaikille tuleville raporteille yhteneväisen ja ammattimaisemman ulkoasun ja täten vahvistamalla asiakkaiden mielikuvia yrityksestä toimisi myös yhtenä markkinoinnin työkaluna.

Projektitkokonaisuuteen kuuluivat eri vaiheet projektin määrittelystä ja suunnittelusta aina sen toteutukseen. Opinnäytetyön kirjallinen osuus painottuu määrittelyvaiheeseen sekä tietokannan ja käyttöliittymän suunnitteluun, koska näissä kokonaisuuksissa kehittyminen ja tämän kehityksen esilletuonti on oman urasuunnitelmani kannalta olennaisempaa kuin ohjelmointitoteutukseen keskittyminen.

Työtä määriteltäessä ja suunniteltaessa tutustuttiin yrityksen nykymalliseen raportointikäytäntöön ja uuteen raporttimalliin saatiinkin monia suuntaviivoja aiempien raporttien sinänsä hyvistä osioista ja ajatusmalleista.

2 OHJELMISTOTUOTANNON KÄYTÄNTÖJÄ

Ohjelmistotuotannolla tarkoitetaan tietokoneohjelmistojen kehittämisessä yleisesti käytettyjä tekniikoita, työkaluja, menettelytapoja ja periaatteita. Vaikka ohjelmistotuotanto onkin käsitteenä ollut käytössä jo vuosikymmeniä, sen kattavuudesta on kiistelty jatkuvasti, koska sen takana ei ole muissa insinööritieteissä yleisesti käytettyjä puhtaammin matemaattisia menetelmiä, vaan se pohjautuu enemmänkin juuri hyväksi havaituille toimintavoille ja periaatteille. Laajasti ajateltuna koko ohjelmistotuotannon ohjaavaksi tavoitteeksi on tuottaa asiakkaiden vaatimusten mukaisten ohjelmistojen tuottamista mahdollisimman hyvin ennustettavilla budjeteilla ja kehitysaikatauluilla. Näin ollen kaikki ohjelmistokehityksen eri vaiheet (muutamina esimerkkeinä määrittely, suunnittelu, toteutus, testaus ja dokumentointi) voi laskea kuuluvaksi ohjelmistotuotannon piiriin (Haikala, I. & Mikkonen, T. 2011, s. 11-12).

Tässä opinnäytetyössä suurin painoarvo on toteutetun projektin määrittely- ja suunnitteluosuuksilla, mistä syystä pyrin myös tässä tutkivassa osuudessa keskittymään eniten juuri näihin projektin kannalta oleellisimpiin käytäntöihin.

2.1 Vaatimusten käsittely (requirements engineering)

Vaatimusten käsittely on yksi ohjelmistokehityksen perustehtävistä. Vaatimukset muodostavat keskeisen perustan ohjelmistokehityksen myöhemmille vaiheille ja sitä käytetäänkin jossain muodossa lähes kaikissa suuremmissa projekteissa. (Paakki, J. 2011, s. 2)

Vaatimukset ovat ominaisuuksia, joita tuotteella pitää olla, tai asioita, joita sillä pitää voida tehdä. Vaatimuskäsittely on yksi onnistuneen ohjelmistoprojektin perusedellytyksistä. Tutkimusten valossa huono vaatimusten käsittely on ollut suurimpia yksittäisiä syitä epäonnistuneiden ohjelmistoprojektien ongelmien takana (Haikala, I. & Mikkonen, T. 2011, s. 61, Paakki J. 2011, s. 5).

Vaatimusten käsittely tuntuu lähteestä riippuen tarkoittavan eri ihmisille hieman eri asioita, koska tapoja sen toteuttamiseen on monia. Joissakin pienemmissä projekteissa vaatimusten käsittelyyn saattaa riittää nenäliinan kulmaan käsin kirjoitettu kuvaus tarpeista, suuremmissa tuotannoissa vaatimusten käsittely saattaa käsittää satojakin sivuja dokumentaatiota, eri sidosryhmien haastatteluita, ideointipalavereita, tai esimerkiksi markkinointitutkimuksia. Kaikki käytetyt lähteet kuitenkin nimeävät myös muutamia yhteisiä piirteitä vaatimusten käsittelyn prosessille, jossa selvitetään, mitä ominaisuuksia ohjelmistolta vaaditaan ja kuinka nämä vaatimukset parhaiten saataisiin kuvatuksi jatkokehitykseen soveltuviksi.

Ohjelmiston vaatimusten käsittely lähtee aina asiakkaan liiketoiminnallisten tavoitteiden ymmärtämisestä (Haikala, I. & Mikkonen, T. 2011,, s. 61). Nämä asiakkaasta lähtevät vaatimukset jaotellaan tavallisesti ohjelmistotuotannon kannalta toiminnallisiin ja ei-toiminnallisiin vaatimuksiin (Paakki, J. 2011, s. 2-4, 26, Pohjonen, R. 2002, s. 28-29, Wikipedia 2014b). Lyhykäisyydessään toiminnallisilla vaatimuksilla kuvataan sitä, mitä järjestelmällä tulee voida tehdä ja ei-toiminnallisilla vaatimuksilla puolestaan enemmän sitä mitä järjestelmä on (Wikipedia 2014c, Wikipedia 2014d).

2.1.1 Toiminnalliset vaatimukset

Toiminnalliset vaatimukset kertovat, kuinka järjestelmä toimii ja kommunikoi käyttäjän ja muiden järjestelmien kanssa (Pohjonen, R. 2002, s. 28, Paakki, J. 2011, s. 27, Wikipedia 2014c). Näitä toiminnallisia vaatimuksia voidaan kuvata esimerkiksi käyttötapausten ("use cases") kautta esimerkiksi UML-mallinnuksen avulla (Wikipedia 2014b). Esimerkkeinä toiminnallisista vaatimuksista käyttäjän tulee esimerkiksi voida lisätä kuvia tuotteen tietosivulle, tai asiakkaan tulee voida tarkastaa omat tilaustietonsa käyttäjätilitään.

2.1.2 Ei-toiminnalliset vaatimukset

Siinä missä toiminnalliset vaatimukset kuvaavat enemmän yksittäisten toimintojen ominaisuuksia, ei-toiminnalliset vaatimukset määrittävät koko ohjelmiston toimintaan ja suunnitteluun vaikuttavia tekijöitä. Ne määrittelevät, minkälaisen reunaehtoisen vallitessa järjestelmä täyttää toiminnalliset vaatimuksensa (Pohjonen, R. 2002, s. 28), tai millä työkaluilla ja menetelmillä kehitystyötä tulee tehdä (Paakki, J. 2011, s. 3, 28)

Ohjelmisto voidaan esimerkiksi haluta toteutettavan Microsoftin Visual Studio-kehitystyökalulla C#-kielellä ja sen on toimittava Microsoft Windows 8-ympäristössä kosketusnäyttösovelluksena. Asiakkaan ohjelmistoon tekemään hakuun on puolestaan saatava tuotettua vastaus 4:ssä sekunnissa. Muita esimerkkejä ei-toiminnallisista vaatimuksista voivat olla varmuuskopiointiin, sertifiointiin, laajennettavuuteen, lähdekoodin avoimuuteen, tietoturvallisuuteen, testattavuuteen tai hinnoitteluun liittyvät vaatimukset (Paakki, J. 2011, s. 30-34, Pohjonen, R. 2002, s. 28-29, Wikipedia 2014b).

2.2 Protoilumalli (software prototyping)

Joskus tekstimuotoinen dokumentointi ei ole paras tapa kuvata kehitettävän ohjelmiston toimintaa asiakkaalle. Tällaisissa tilanteissa ohjelmiston prototyypitys voi olla suureksi hyödyksi (Paakki, J. 2011, s. 64). Prototyypityksellä tarkoitetaan joillain tavoin vaillinaisen, mutta kuitenkin oikean ohjelman näköisen version rakentamista eri ominaisuuksien määrittelemisen ja tutkimisen helpottamiseksi. Esimerkkejä tutkittavista seikoista voivat olla käyttöliittymälogiikka, uusi teknologia tai vaikkapa ohjelmiston suorituskyvylliset yksityiskohdat. Parhaimmillaan prototyypillä voidaan kartoittaa monia, aiemmin huomaamattomia ohjelmiston toiminnallisia vaatimuksia, mutta ei-toiminnallisten vaatimusten kartoittamisessa niistä ei menetelmän luonteen vuoksi usein kuitenkaan ole mainittavaa hyötyä (Paakki, J. 2011, s. 64).

Protoilumallissa käytetään poisheitettäviä prototyyppejä, evoluutioprototyyppejä tai tarpeen mukaan näiden kahden yhdistelmiä (Haikala, I. & Mikkonen, T. 2011, s. 38).

2.2.1 Poisheitettävät prototyypit (throwaway prototypes)

Poisheitettäviä prototyyppejä käytetään usein käyttöliittymälogiikan suunnitteluun ja demoamiseen. Tämä voi yksinkertaisimmillaan tarkoittaa paperille piirrettyjä kuvia, mutta prototyyppi voi myös olla navigaatioltaan toimiva ja jo oikeitakin näyttöjä ja toimintalogiikkaa sisältävä malliohjelmisto (Haikala, I. & Mikkonen, T. 2011, s. 39).

Poisheitettävät prototyypit eivät nimensä mukaisesti luo pohjaa lopulliselle sovellukselle, vaan niitä käytetään ensisijaisesti vaatimusmäärittelyn ja suunnittelutyön aputyökaluina, joilla asiakkaalle voidaan nopeasti tuottaa alustava näkymä siitä, miltä projektissa asetetut vaatimukset voisivat näyttää lopullisessa tuotteessa.

Tämä menetelmä mahdollistaa nopealla toteutettavuudellaan projektien alkuvaiheissa vastaan tulevien, lopulliselle sovellukselle kriittisten määrittely- ja suunnitteluvaiheiden tarkentamisen. Nämä tarkennukset saattavat johtaa suuriin rahallisiin ja ajallisiin säästöihin, kun jo tehtyä työtä ei tarvitse väärinkäsitysten tai vaatimusmäärittelyn epätasällisyyksien takia lähteä korjaamaan, tai aloittamaan alusta.

Sovellusmuotoon tuotettuna prototyypin käyttöliittymähahmotelma myös auttaa asiakasta paremmin hahmottamaan ohjelmiston toiminnan myös käytännön tasolla. Näin asiakkaan on luontaisesti helpompi jo varhaisessa vaiheessa nostaa esiin huomioitaan mahdollisista visuaalisista muutoksista, tai vaihtoehtoista toteutustavoista käyttöliittymälogiikassa.

Poisheitettävät prototyypit eivät aina kuitenkaan ole edes ohjelmamuotoon asti tuotettuja hahmotelmia. Yksinkertaisimmillaan prototyyppi voidaankin toteuttaa jo pelkällä kynällä ja paperilla. (Wikipedia 2014a).

2.2.2 Evoluutioprototyypit (evolutionary prototypes)

Evolutiiviset prototyypit pyritään järjestelmällisesti rakentamaan toimivammiksi kuin poisheitettävät prototyypit. Tämän menetelmän ajatus on aloittaa niistä vaatimuksista ja ominaisuuksista jotka jo tiedetään ja tunnetaan varmoiksi, minkä jälkeen prototyyppiä laajennetaan ja muokataan askel askeleelta lähemmäksi tuotantoversiota. Evolutiivinen prototyyppi muodostaakin näin ohjelmiston ytimen, jonka ympärille lisätään toiminnallisuutta sitä myöten, kun eri osioiden vaatimuksia saadaan tarkemmin määriteltä.

Evolutiivinen prototyyppi saattaa asiakkaan näkökulmasta olla tietyssä mielessä tehokkaampi ja käytännöllisempi ratkaisu, kuin poisheitettävä prototyyppi, koska asiakkaalle voidaan jo alusta asti tarjota käyttöön jo jonkinasteisesti toimiva ohjelmisto.

Evolutiivisessa protoamisessa kehittäjätiimi keskittyy ohjelmistossa pienempiin osioihin kerrallaan sen sijaan, että kehittäisi koko ohjelmistoa samanaikaisesti. Tämä toimii riskinhallintamekanismina. Jos joidenkin osioiden vaatimusmäärittelyt ovat vielä epäselviä tai kesken, tai kehittäjältä puuttuu tiettyjen ominaisuuksien toteuttamiseen vaadittava tietotaito, on usein turvallisempaa suunnata huomio ennemmin niihin osioihin, jotka voidaan tuottaa nopeammalla aikataululla ja pienemmällä myöhempien, perustavanlaatuisien muutosten suhteen pienemmällä riskillä (Wikipedia 2014a).

2.2.3 Protoilumallin haasteet

Protoilumallin käyttäminen vaatii ohjelmistokehittäjältä järjestelmällisyyttä ja pitkäjänteisyyttä, jotta sitä saataisiin hyödynnettyä optimaalisesti. Esimerkiksi testausvaiheessa saattaa yhä jäädä huonosti toimivia tai rikkiinäisiä osuuksia osaksi lopullista ohjelmistoa, totuuden paljastuessa vasta kun koko ohjelmisto otetaan käyttöön.

Prototyyppien toinen selkeä ongelma on niiden synnyttämä hieman valheellinen kuva ohjelmiston tilanteesta. Asiakas voi herkästi kuvitella, että ohjelmisto on jo lähellä julkaisuvalmiutta, vaikka siitä todellisuudessa olisikin saatu toteutettua vasta pintapuolinen käyttöliittymä ilman ohjelmakoodia. Tästä syystä voidaankin suositella, että prototyyppi näyttäisi ja tuntuisi tarkoituksella keskeneräiseltä (Haikala, I. & Mikkonen, T. 2011., s. 38-39).

On myös ensiarvoisen tärkeää, ettei prototyyppeihin (etenkään poisheitettäviin) uhrata liikaa aikaa ja resursseja. Mallin kantava ajatus on päästä toteuttamaan hahmotelmia nopeasti, minkä jälkeen siitä on tarkoitus siirtyä eteenpäin. Liian yksityiskohtaisen prototyypin kehittäminen saattaa nopeasti harhauttaa kehittäjää keskittymään epäolennaisuuksiin. On myös tärkeää huomioida, ettei kaikkia prototyypin ominaisuuksia ole järkevää toteuttaa lopulliseen tuotteeseen, vaikka ne kehittäjän tai asiakkaan mielestä saattaisivatkin olla hyödyllisiä (Wikipedia 2014a).

3 CASE: TIETOKANTAPOHJAISEN RAPORTINHALLINTAJÄRJESTELMÄN MÄÄRITTELY JA SUUNNITTELU ASIAKKAALLE

Opinnäytetyönä raportinhallintajärjestelmän tilannut asiakas on 1990-luvulla perustettu kattourakointeihin ja erilaisiin vesieristykseen erikoistunut perheyritys. Yritys urakoi Lounais- ja Etelä-Suomen alueilla, minkä lisäksi se myy materiaaleja koko maan laajuisesti. Toimipisteet yrityksellä on Salossa ja Vantaalla.

3.1 Tarve uudelle raportinhallintajärjestelmälle

Yrityksen tarjoamiin palveluihin kuuluvat mm. monenlaiset eri kattohuollot ja -työt, eristystyöt, pihakannet, sekä sokkelit. Koska yritys tarjoaa monenlaisia palveluita ja asiakkaiden tarpeet ovat tapauskohtaisia ja suuresti toisistaan eroavia on kulloisenkin kohteen tarkastus, tarvittavien toimenpiteiden suunnittelu ja kustannusarvioiden laskeminen asiakkaalle harkittavaksi ensiarvoisen tärkeää.

Yritys on valmistellut näistä tarkastuksista tehtäviä raportteja vaihtelevin työkaluin ja menetelmin, mistä syystä lopputuotteet ovat olleet sisällöltään ja ulkoasultaan suuresti eroavia. Tämän toimintamallin huonoihin puoliin kuuluu myös sen suuri ajankulutus, yksittäinen työntekijä on saattanut joutua käyttämään työpäivän verran työtunteja yksittäisen raportin kirjoittamiseen ja asemointiin tulostamista varten.

Tehtävänäni oli siis tämän ongelmakohdan korjaamiseksi suunnitella ja toteuttaa yrityksen käyttöön tietokantapohjainen raportinhallintajärjestelmä, jolla työvaiheen syömiä työtunteja saataisiin olennaisesti vähennettyä ja joka yhdenmukaistaisi raporttien sisällön ja ulkoasun ammattimaisemmiksi.

3.2 Raportinhallintajärjestelmän määrittely

3.2.1 Alkupalaveri

Projektin aloituspalaveri pidettiin 25.1.2013 Turun ammattikorkeakoulun Salon toimipisteen tiloissa. Palaveriin osallistuivat itseni lisäksi opinnäytetyöni ohjaaja Päivi Killström ja toimeksiantajan yhteyshenkilönä projektissa toiminut konsultti.

Yhteyshenkilö oli tuonut mukanaan muutamia vanhoilla menetelmillä tuotettuja raportteja, joista palaverin aikana yhteistyössä lähdimme muodostamaan kuvaa siitä, mitä sisältöä ja millaisessa paketissa tulevan järjestelmän raporteista tulisi löytymään.

Tuotettava ohjelmiston tulisi mahdollistaa halutunlaisten raporttien helppo luominen syöttämällä graafiseen tietokannan päällä pyörivään käyttöliittymään tarvittavat tiedot. Lisäksi ohjelmistosta tulee voida hakea jo aiemmin luotuja raportteja jatkotyöstöä tai tarkastelua varten. Raportteja tulee voida tulostaa asiakkaalle paperisina ja sähköisinä versioina.

Käyttöönottoaikatauluksi tässä alkupalaverissa suunniteltiin testausvaiheen alkamista kesällä 2013 ja tuotantoversion julkistamista syyskuksi 2013.

3.2.2 Käyttöliittymän prototyyppi

Toimeksiantajan todellisten tarpeiden tarkempaa määrittelyä ajatellen aloitin alkupalaverissa käydyn keskustelun ja malliksi annettujen aiemmin tehtyjen raporttien perusteella käyttöliittymäprototyypin hahmottelun Microsoft Visual Studiolla. Tällä poisheitettävällä prototyypillä oli tarkoitus luonnollisesti hahmotella ohjelmiston käyttäjäkokemusta ja raporttien työstämisprosessia, sekä tarkennuksia uuden raporttimallin sisältöön ja ulkoasuun. Näillä tarkennuksilla haettiin apua myös tietokannan hahmotteluun, missä kerätty lisätieto tulikin lopulta auttamaan suuresti. Eri tietojen kytkeytyminen kannassa toisiinsa ja se, miten ja missä eri tietoja tulitaisiin ohjelmassa käyttämään auttoi

suuresti kun hahmottelusta siirryttiin kantaa toden teolla suunnittelemaan tarvetta vastaavaksi.

Visual Studio tarjoaa skaalautuvuutensa ansiosta helppokäyttöisiä työkaluja mm. käyttöliittymien ja yksinkertaisempien perustoimintojen nopeaan toteuttamiseen. Se oli myös itselleni jo tuttu kehitysympäristönä, mistä syistä prototyyppi oli mahdollista toteuttaa nopealla aikataululla ilman sen enempää opiskelutarvetta.

Alkuperäisenä ajatuksenani oli myös tuotantoversion toteuttaminen Visual Studion ja Accessin yhteisapelillä juuri aiemman osaamisen tuomien hyötyjen vuoksi. Tuotantoversiota päädyttiin kuitenkin lopulta kehittämään kokonaisuudessaan Access-ympäristöön toimeksiantajan ja ohjaavan opettajan aiempien hyvien kokemusten pohjalta. Päätös olikin teknisesti perusteltu, Access työkaluna kun sopii ominaisuuksiltaan kyseisenkaltaiseen toteutukseen sujuvasti. Prototyypillä saatiinkin jo näin varhaisessa vaiheessa tarkennettua ja varmistettua käytettävä kehitysympäristö, johon asiaan ei aiempien vaatimusmäärittelykokousten aikana oltu kiinnitetty huomiota.

Prototyypin käyttöliittymän rakenne oli toimeksiantajan mielestä selkeä ja sinänsä lupaava, mistä syystä sen logiikkaa käytettiin pohjana tuotantoversion kehityksen alkaessa. Hyvästä pohjasta huolimatta tuotantoversioon rakennettu käyttöliittymä koki myös luonnollisesti muutoksia kehitysprosessin aikana. Erinäisiä toimintoja pyrittiin asemoimaan ja toteuttamaan paremmin ja niiden käyttöä pyrittiin selkeyttämään entisestään.

Prototyyppiä esiteltiin asiakkaalle sekä kirjallisesti sähköpostin välityksellä (Liite 1: Prototyypin esittely asiakkaalle), että asiakkaan itsensä nähden tosielämän demona suunnittelutapaamisen aikana.

3.2.3 Raportinhallintajärjestelmän vaatimukset

Liiketaloudellinen lähtökohta raportinhallintajärjestelmän kehitykselle on asiakkaan tarve saada tehostettua raportointiprosessiaan. Nykyiset käytännöt

haaskaavat työtunteja ja tuottavat sekä visuaalisesti, että sisällöllisesti epäammattimaista jälkeä. Toteutettavan työkalun tulee tehostaa yrityksen työntekijöiden työajan käyttöä helpottamalla raportointia ja tuottaa parempaa markkinointia visuaalisesti ja sisällöllisesti ammattimaisempien asiakasraporttien muodossa.

Suunnittelupalavereiden ja määrittelyn aputyökaluna käytetyn, poisheitettävän prototyypin kautta saimme yhteistyössä asiakkaan kanssa rajattua tarkemmin kehitettävältä työkalulta vaadittuja ominaisuuksia.

Ohjelmisto tulee toteuttaa Microsoft Windows XP/7/8-ympäristöön ja se tulee rakentaa Microsoft Access-relaatiotietokantaohjelmistolla, koska asiakkaan itsensä käytössä on Microsoftin Windows-käyttöjärjestelmillä varustettuja tietokoneita ja heillä on jo kokemuksia Accessista työkaluna. Ohjelman rakentaminen Access-työkalulla määrittyi vaatimukseksi vasta prototyypivaiheen kohdalla, aiemmissa suunnittelutapaamisissa toteutustyökalua ei käsitelty.

Ohjelmistoon tulee voida lisätä raporteissa tarvittavat tiedot ja ohjelmasta tulee voida tulostaa ulos valmiita raporttikokonaisuuksia sekä paperisina, että sähköisesti PDF-versioina. Käyttäjän tulee voida valita, mitä tietoja raportin tarkastusraportti-osioon tulostetaan. Vaikka raporttien asiasisällössä esiintyykin vaihtelua arvioitujen kohteiden perusteella, on kaikkien raporttien noudatettava kuitenkin yhtenäistä, määriteltyä visuaalista linjaa ammattimaisemman vaikutelman ja selkeyden luomiseksi.

Ohjelmiston on tunnistettava käyttäjä tämän Windows-käyttäjä tunnuksen perusteella. Tietokannan ylläpitoa ja vakioarvojen muokkaamista on voitava rajata vain halutuille käyttäjille. Yrityksen työntekijöiden tulee voida päästä käsiksi ohjelmiston tietokantaan kaikilta asiakkaan eri kaupungeissa sijaitsevilta toimipisteiltä.

Ohjelmisto tulisi saada testikäyttöön kesällä 2013 ja tuotantoon syyskuksi 2013.

3.2.4 Käyttötapaukset (use cases)

Käyttötapaukset (use cases) ovat tapa kuvata käyttäjän ja sovelluksen välistä interaktiota. Käyttötapausten avulla kerrotaan, mitä eri asioita ohjelmistolla voidaan tehdä. Niiden on tarkoitus kuvata yksittäisiä toimintoja mahdollisimman tarkasti ja selkeästi (Niemi, O. 2013, s. 17). Olen opinnäytetyöhöni koonnut muutamia ohjelman perustoimintoja kuvaavia käyttötapauksia.

Ensimmäinen käyttötapaus on taulukossa 1 kuvattu uuden raportin lisääminen. Kyseinen toiminto on jo kuten kuvauksensa kertoo, olennainen osa ohjelmiston päivittäistä käyttöä, mistä syystä prosessin on oltava virtaviivainen ja selkeä.

Taulukko 1. Uuden raportin lisääminen.

Nimi	Uuden raportin lisääminen
Suorittaja	Käyttäjä
Esiehdot	<ol style="list-style-type: none"> 1. Uusi raportti pitää lisätä kantaan. 2. Käyttäjällä on tarvittavat tiedot raportin lisäämistä varten.
Kuvaus	<ol style="list-style-type: none"> 1. Käyttäjä painaa etusivun Uusi raportti-painiketta. 2. Käyttäjä syöttää raportin nimen, ja tarkastaa että raportin syöttöpäivämäärä on oikein. 3. Käyttäjä tarkastaa, että ohjelma on tuonut oikeat käyttäjätiedot raportin kirjoittajan tietoihin ja tarvittaessa hakee oikeat tiedot Hae käyttäjää-painikkeella. 4. Käyttäjä tallentaa uuden raportin Tallenna-painikkeella.
Poikkeukset	<ol style="list-style-type: none"> 1. Uutta raporttia ei sittenkään haluta lisätä. 2. Ohjelma ei sisällä käyttäjän tietoja raportin kirjoittajan tietoihin, tai antaa väärää tietoja.
Lopputulokset	Uusi raportti lisätään järjestelmään ja se sen muut osiot ovat nyt muokattavissa.

Mikäli uutta raporttia ei sittenkään haluta lisätä järjestelmään poikkeuksen 1 mukaisesti, voi käyttäjä perua toimenpiteen painamalla Sulje tallentamattapainiketta. Mikäli ohjelmisto tarjoaa poikkeuksen 2 mukaisesti väärää käyttäjätietoja, eikä käyttäjä saa Hae käyttäjää-painikkeella haettua omia tietojaan esille, tulee hänen ottaa yhteys kantaa ylläpitävään henkilöön. Tämä virhe voi johtua joko siitä, ettei kyseisen käyttäjän tietoja ole vielä lisätty kantaan, hänen nimensä ja/tai yhteystietonsa ovat muuttuneet mutta niitä ei ole päivitetty kantaan, tai hänen Windows-kirjautumistunnuksensa on muuttunut eikä sitä ole päivitetty kantaan.

Toinen käyttötapaus on taulukossa 2 esitetty jo luodun raportin hakeminen tarkasteltavaksi, muokattavaksi tai tulostettavaksi. Kuten uuden raportin lisääminen, myös jo luotujen raporttien hakeminen jatkotoimenpiteitä varten on tärkeä osa ohjelman jokapäiväistä käyttöä.

Taulukko 2. Raportin hakeminen ohjelmistosta.

Nimi	Raportin hakeminen ohjelmistosta
Suorittaja	Käyttäjä
Esiehdot	<ol style="list-style-type: none"> 1. Jo luotu raportti halutaan etsiä kannasta tarkasteltavaksi, muokattavaksi tai tulostettavaksi. 2. Käyttäjällä on tarvittavat tiedot haun rajaamiseksi.
Kuvaus	<ol style="list-style-type: none"> 1. Käyttäjä painaa etusivun Hae raporttia-painiketta. 2. Käyttäjä syöttää hakuikkunasta löytyviin hakuehtoihin arvot, mikäli hakua tarvitsee rajata. 3. Käyttäjä painaa Hae-nappia. 4. Käyttäjä valitsee haun tuottamasta listasta halutun raportin ja painaa joko Tulosta valittu-painiketta (halutessaan tulostaa raportin) tai Muokkaa valittua-painiketta (halutessaan tarkastella tai muokata raporttia).
Poikkeukset	<ol style="list-style-type: none"> 1. Hakua ei halutakaan suorittaa. 2. Etsittävää raporttia ei olekaan syötetty järjestelmään.

	3. Hakuehdot eivät toimi
Lopputulos	Etsitty raportti joko tulostetaan tai se avataan muokattavaksi.

Mikäli hakua ei sittenkään haluta poikkeuksen 1 mukaisesti suorittaa, pääsee käyttäjä takaisin ohjelmiston aloitusnäkyeseen sulkemalla hakuikkunan. Mikäli etsittävää raporttia ei olekaan vielä syötetty järjestelmään, voi käyttäjä sen lisätä käyttäjätapauksen 1 mukaisesti. Mikäli hakuehdot eivät tuota haluttuja tuloksia, käyttäjän kannattaa tarkastaa syöttämänsä arvot virheiden varalta. Mikäli hakuehdot eivät kuitenkaan sitten teknisistä syistä eli ohjelmointivirheistä johtuen onnistu rajaamaan hakua oikein, tulee käyttäjän ottaa yhteyttä ohjelmiston pääkäyttäjään, joka osaa ohjata korjaustarpeen eteenpäin, mikäli ei itse ongelmaa voi korjata.

Kolmas tähän poimimani käyttötapaus on toimenpiteiden kustannusarvioiden lisääminen raportin tietoihin. Nämä kustannusarviot tulostuvat raportin yhteenveto-sivulle (Liite 3, s. 4) korjaustarpeen kiireellisyyden mukaisesti värikoodattuna.

Taulukko 3. Toimenpiteiden kustannusarvioiden lisääminen.

Nimi	Toimenpiteiden kustannusarvioiden lisääminen
Suorittaja	Käyttäjä
Esiehdot	<ol style="list-style-type: none"> 1. Raportti on lisätty järjestelmään. 2. Käyttäjällä on tarvittavat tiedot kustannusarvioista.
Kuvaus	<ol style="list-style-type: none"> 1. Käyttäjä joko lisää uuden raportin tai hakee jo luodun raportin muokattavaksi. 2. Käyttäjä avaa kustannusarvioiden lisäysikkunan painamalla Kustannusarvio-painiketta. 3. Lisäysikkunassa käyttäjä painaa Lisää uusi-painiketta lisätäkseen uuden tietueen. 4. Käyttäjä syöttää korjauskohteen nimikkeen, valitsee listasta korjausaikataulun ja syöttää hinta-arvion.

	5. Käyttäjä tallentaa syötetyn tiedon Tallenna ja sulje-painikkeella.
Poikkeukset	<ol style="list-style-type: none"> 1. Uutta tietuetta ei halutakaan lisätä. 2. Lomakkeeseen on syötetty teknisesti virheellistä tietoa. 3. Käyttäjä tallentaa teknisesti oikeanlaista, mutta sisällöltään väärää tietoa.
Lopputulokset	Raportin tietoihin lisätään tietue kustannusarviosta.

Mikäli uutta tietoa ei poikkeustilanteen 1 mukaisesti haluta sittenkään lisätä, pääsee käyttäjä peruuttamaan toimenpiteen Poistu tallentamatta-painikkeella. Mikäli lomakkeelle koitetaan poikkeuksen 2 mukaisesti syöttää teknisesti virheellistä tietoa, tulee ohjelman ilmoittaa tästä käyttäjälle ja estää kyseisten tietojen tallennus. Mikäli käyttäjä toisaalta tallentaa lomakkeelle teknisesti oikeanlaista, mutta sisällöltään väärää tietoa poikkeuksen 3 mukaisesti, voi hän korjata virheen itse. Palattuaan tallennuksen jälkeen kustannusarvioiden lisäysikkunaan käyttäjä voi valita raportille lisätyistä kustannusarvotiedoista haluamansa ja joko poistaa nämä Poista valittu-painikkeella tai muokata näitä Muokkaa valittu-painikkeella.

3.3 Raportinhallintajärjestelmän suunnittelu

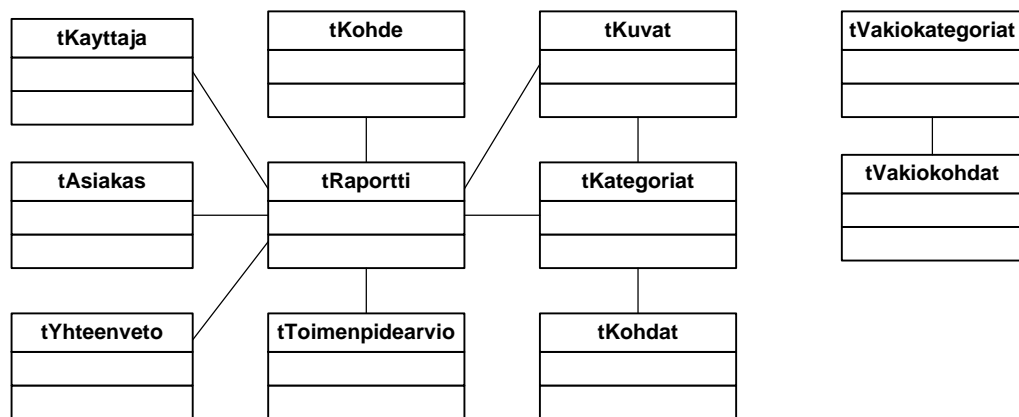
Ohjelmiston eri kehitysversioiden välillä tapahtuneet muutokset saattavat esitystavasta riippuen olla projektin ulkopuoliselle henkilölle vaikeasti hahmotettavia. Kehitysaskelien hahmottamista helpottaakseni pyrin käsittelemään ohjelmiston eri osia pala kerrallaan, jotta lukijalle saataisiin muodostettua selkeämpiä vertailukohtia viimeisimmästä versiosta aiempiin. Olen tähän kirjoitettuun raporttiini valinnut tilan ja ajan säästämiseksi kolme evolutiivisena prototyyppinä työstetyn ohjelmiston eri kehitysvaihetta, joista vertailua suoritan. Ensimmäinen versio on "0.1.0" eli ensimmäinen asiakkaalle esitelty versio määrittelyssä käytetyn, poisheitetyn prototyypin jälkeen.

Välivaiheeksi valitsin version "0.6.0", josta löytyy jo selkeitä kehitysaskela ensimmäiseen versioon, mutta joka on kuitenkin vielä viimeistä toimitettua versiota rajatumpi toiminnaltaan. Viimeisin asiakkaalle toimitettu versio "0.9.0" onkin jo käyttöönottoa ajatellen pienempiä korjauksia vajaa valmis tuote.

Kehitysverailuun olen valinnut kolme ohjelmiston eri osiota, tietokannan, käyttöliittymän- ja logiikan, sekä ohjelmiston lopputuotteen, eli asiakkaalle tulostuvan raportin. Projekti toteutettiin evolutiivisen prototyypin menetelmin, mistä syystä kaikki nämä kolme osiota ovat läpikäyneet sekä suunnittelullisia, että toteutuksellisia muutoksia projektin aikana. Uskonkin, että näiden kolmen osion eri kehitysvaiheiden vertailut tuottavat varsin kattavan kuvan ohjelmiston laajemmasta kehityskaaresta.

3.3.1 Tietokanta

Ohjelmiston tietokannan toteutustyökaluksia varmistui jo projektin aloituspalaverissa Microsoft Access-tietokantaohjelmisto asiakkaan aiempien hyvien kokemusten perusteella. Jotta asiakkaalle olisi mahdollisimman nopeasti voitu esitellä alustava versio lopputuotteesta, eli kannan tietojen perusteella tulostuvasta raportista, lähdettiin myös tietokantaa suunnittelemaan heti tuotantoversion kehityksen alkaessa. Tämä kuvassa 1 esitetty kehityksen alussa suunniteltu kannan perusrakenne pysyi kohtuullisen muuttumattomana läpi kehityskaaren. Suurimmat kantaan tehdyt muutokset kohdistuivat teknisen toteutuksen parantamiseen tietokantaa normalisoimalla. Normalisointi on tietokannan tietojen järjestämisen prosessi, jossa taulukoita, tietueita ja yhteyksiä organisoidaan tietokannan sisältämien tietojen suojaamiseksi ja redundanssin eli saman tiedon useasti tallentamisen välttämiseksi (Microsoft 2014).



Kuva 1. Tietokannan rakenne.

Kannan perusrakenteeseen määriteltiin taulut "tRaportti", "tKayttaja", "tAsiakas", "tYhteenveto", "tKohde", "tToimenpidearvio", "tKategoriat", "tKohdat", "tKuvat", "tVakiokategoriat" ja "tVakiokohdat".

Tauluun "tRaportti" tallennetaan raporttia itseään koskevat tiedot eli raportin tunnistenumero, raportin nimike, luontipäivämäärä, sekä raportin kansikuva. Tähän päätauluun linkitetään tietoja tauluista "tKayttaja", "tAsiakas", "tYhteenveto", "tKategoriat", "tKohde" ja "tToimenpidearvio".

Koska yksi käyttäjä voi luoda monia eri raportteja, on järkevää säilöä käyttäjien tiedot omassa taulussaan, sen sijaan että ne syötettäisiin aina raporttikohtaisesti tauluun "tRaportti". Sama logiikka pätee myös asiakkaiden ja kohteiden kohdalla. Samasta kohteesta voidaan tehdä useita raportteja ja sama asiakas voi tilata useita eri arviointeja. Näin ollen myös nämä tiedot kannattaa tallentaa omiin tauluihinsa, redundanssin välttämiseksi.

Taulujen "tYhteenveto" ja "tToimenpidearvio" sisältämien tietojen käsittely on tekniseltä toteutukseltaan ja kannan selkeyden kannalta järkevintä eriyttää myös omiin tauluihinsa. Näistä tauluista linkitetään oikeat tiedot oikeaan raporttiin raporttien tunnistenumeroilla.

Koska jokaiselle raportille tulee voida määritellä raportin tarkastusraportti-osioon (Liite 3, s. 4-5) vain tarpeelliset kohdat, irtautettiin kategoriat ja näiden alakohdat

omiksi tauluikseen. Näihin tauluihin haetaan käyttäjän valintojen perusteella arvot tauluista "tVakiokategoriat" ja "tVakiokohdat" uutta raporttia luodessa.

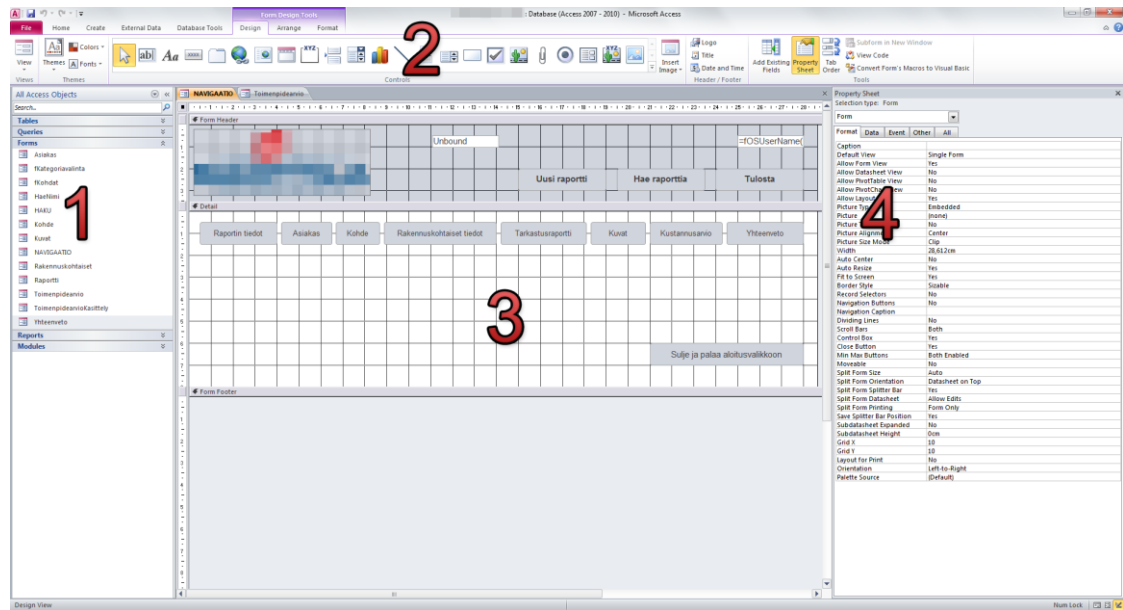
Raporttiin tulostuvat kuvat pitää saada sekä linkittymään oikeaan raporttiin, että raporttia tulostettaessa oikean kategorian alle (Liite 3, s. 6-9), mistä syystä niihin määriteltiin relaatiot sekä tauluun "tRaportti", että "tKategoriat".

3.3.2 Käyttöliittymä- ja logiikka

Uskon itse käytettävyyssasioissa yksinkertaiseen esitystapaan ja ohjelmiston käyttöliittymää suunnitellessani pyrinkin totuttuun tapaan tarjoamaan käyttäjälle näkyviin aina vain ne toiminnot ja tiedot, joita hän käsittelemässään kohdassa tarvitsee. Tämä käyttöliittymän visuaalinen ja toiminnallinen yksinkertaistaminen sai asiakkaalta jo määrittelyvaiheessa luodun poisheitettävän prototyypin perusteella hyvää palautetta ja sitä pidettiin tuotantoversion kehitystä ajatellen varsin lupaavana.

Määrittelyvaiheen prototyypillä saimme asiakkaan kanssa määriteltyä käyttöliittymän toteutustyökaluksi MS Accessin erillisen sovelluksen sijasta. Access tarjoaa useita eri työkaluja kannasta löytyvän tiedon esittämiseen käyttöliittymämuodossa. Käyttöliittymän navigaatio ja muut toiminnallisuudet rakennetaan Microsoftin Visual Studio-kehitysympäristöstäkin tuttujen lomakkeiden ("forms") päälle.

Kuvassa 2 on numerolla 1 osoitettu Accessin itsensä käyttöliittymästä ohjelmistoon lisättyjen lomakkeiden sijainti. Numerolla 2 kuvaan on merkitty Accessin tarjoamia työkaluja. Accessiin on sisäänrakennettu useita erilaisia navigaatio- ja valikkoelementtejä, kuvia, graafeja ja asemointityökaluja. Kuvaesimerkissä muokattavaksi avattuna on numerolla 3 merkitty ohjelmiston etusivu, josta käsin käyttäjä navigoi täyttämään raportin alakohtia. Numerolla 4 kuvaan on merkitty lomakkeelle lisätyille objekteille löytyvät ulkoasu- ja toiminta-asetukset.



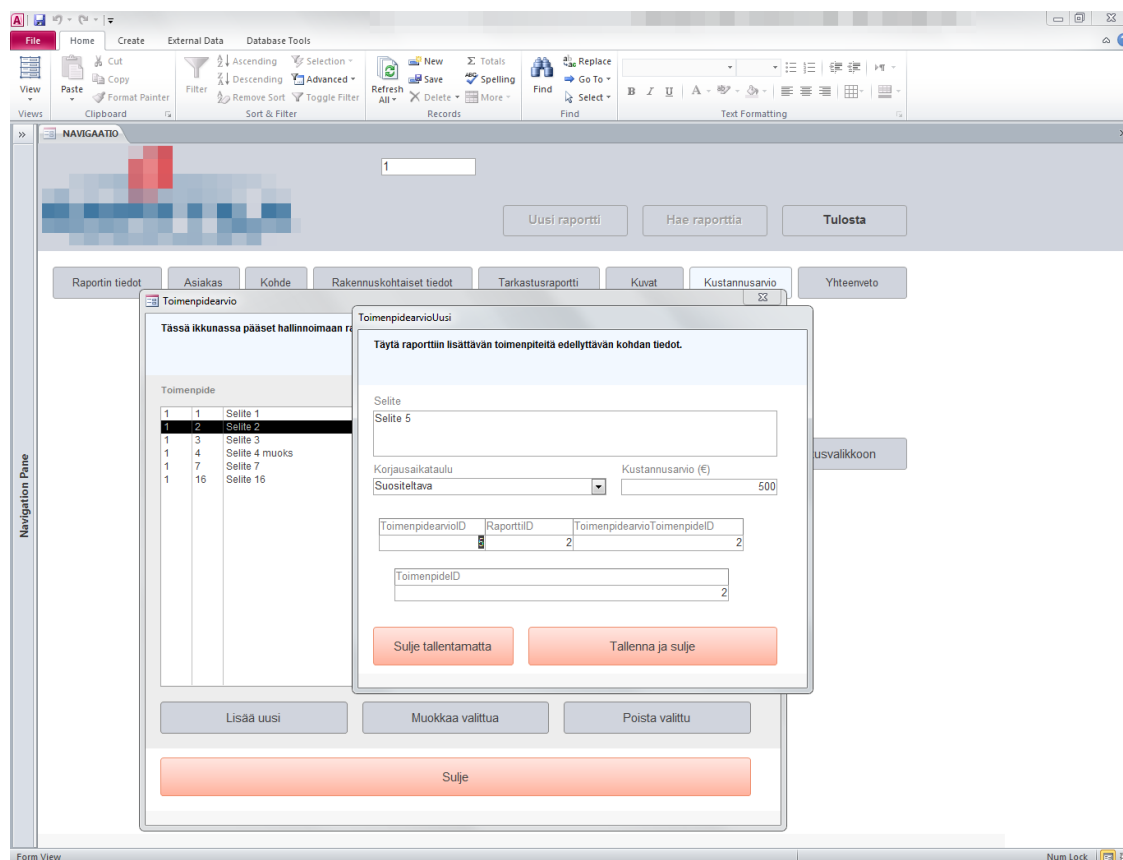
Kuva 2. Accessin työkalut käyttöliittymän rakentamiseen.

Käyttöliittymän rakentamisessa käytettävien lomakkeiden voi ajatella olevan omia sovellussivujaan, joille voidaan puolestaan rakentaa haluttua sisältöä. Näitä sovellussivuja voidaan sisällyttää toisiin sovellussivuihin erilaisten navigaatio- ja sisältöelementtien avulla.

Projektin alkuvaiheessa suurin painoarvo kehitystyössä oli tietokannan ja tulostettavan raportin suunnittelussa, käyttöliittymän osalta hain vielä parasta toteutustapaa ensimmäisen prototyypin mukaisen ulkoasun ja käyttölogiikan rakentamiseksi.

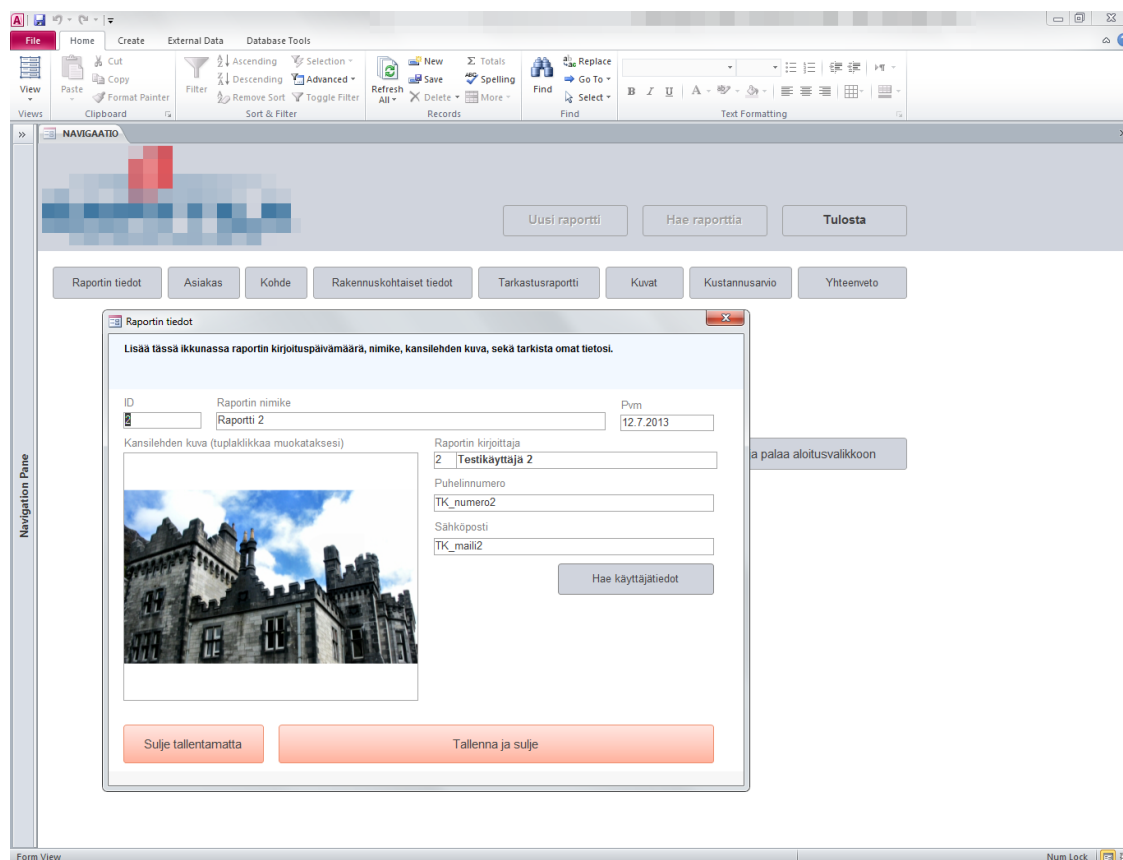
Kuva 3. Käyttöliittymäkuva versiosta 0.1.0.

Kuvassa 3 esitelty version 0.1.0 käyttöliittymä rakentui Accessista löytyvän valmiin navigaatioelementin varaan. Kunkin raportin osa-alue oli toteutettu omana lomakkeenaan, jotka lomakkeet sitten lisättiin päälomakkeen navigaatioelementtiin. Alalomakkeilta löytyvät tiedot olivat jo näennäisen vapaasti muokattavissa muokattavissa, mutta valittu navigaatoratkaisu ei tuntunut toimivalta. Tässä kehitysversiona ei vielä onnistunut uusien raporttien aloittaminen, jo tehtyjen raporttien hakeminen hakukriteerein, eikä raporttien tulostaminen. Toteutustavasta luovuttiin pikaisesti seuraaviin versioihin siirryttäessä vaihtoehtoisen, sujuvammin toimineen toteutustavan löydyttyä.



Kuva 4. Käyttöliittymäkuva versiosta 0.6.0.

Versioon 0.6.0 mennessä olin jo löytänyt toimivan tavan toteuttaa ohjelmassa navigointi ja raportin tietojen rajaaminen ja käsittely. Tätä toimivampaa tapaa esittelin ensin asiakkaalle kehityspalaverissa ja sainkin sille hyväksynnän. Navigaatio lomakkeelta toiselle Accessin navigaatioelementtiä käyttäen korvattiin painonappuloilla, jotka painettaessa avaavat aina halutun raportin osion muokattavaksi omaan ikkunaansa (kuva 4). Käyttöliittymän ulkonäkö ja logiikka saatiin tällä muutoksella muistuttamaan enemmän määrittelyvaiheen prototyyppiä.



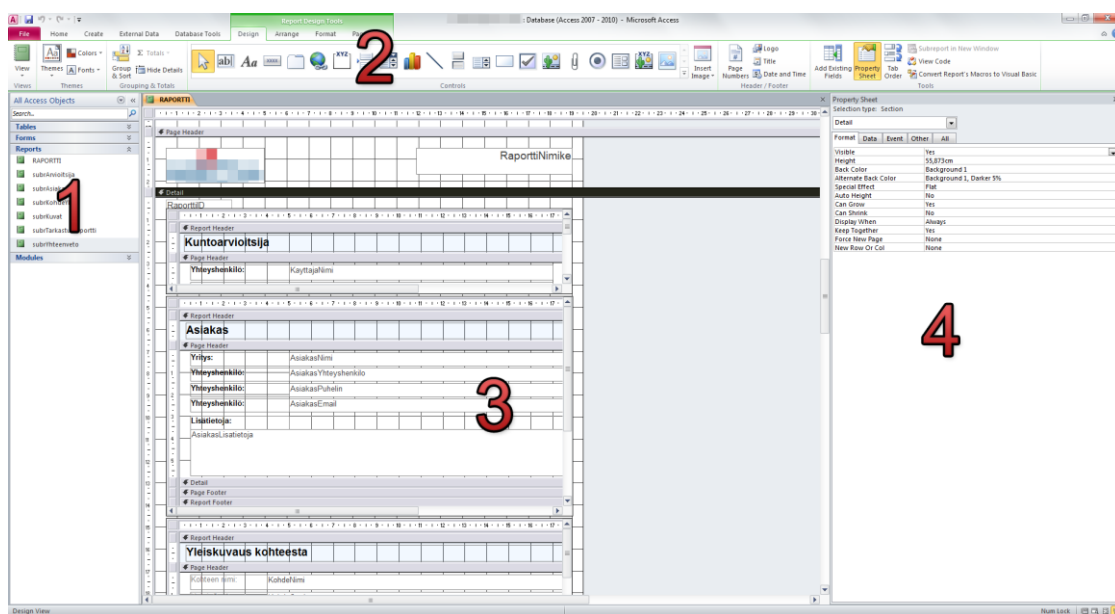
Kuva 5. Käyttöliittymäkuva versiosta 0.9.0.

Vaikka versio 0.6.0 rupesikin jo käyttöliittymältään ja –logiikaltaan olemaan lähellä lopullista versiotaan, oli siinä kuitenkin vielä yksityiskohtia hiottavana., esimerkiksi raportin muutamien osioiden alalomakkeilla oli vielä parannettavaa sisällöllisesti ja visuaalisesti. Lisäksi kehitysversioissa käytössä olleet erilaiset tarkastusnumerot ja muut lähinnä kehittäjän testatessa työtä helpottavat apuvälineet karsittiin pois asiakkaalle lähetettävistä myöhemmistä versioista (kuva 5). Suurin osa versioiden 0.6.0 ja 0.9.0 käyttöliittymien välillä tapahtuneista kehitystoimenpiteistä kohdistuikin pinnan alta löytyvään ohjelmointitoteutukseen, ulkoisesti suurimmat loppukäyttäjälle näkyvät erot rajoituivat pitkälti juurikin lomakkeiden siistimiseen.

3.3.3 Raportti

Asiakkaalla oli jo projektin alkupalaverissa mukanaan muutamia yrityksen aikaisemmin tuottamia asiakasraportteja. Mukaan oli valikoitu sellaisia raportteja, joilla koettiin olevan hyviä ulkoisia tai sisällöllisiä piirteitä ja joita voitaisiin käyttää pohjana uuden raporttimallin suunnittelussa. Jo alkupalaverissa saimme varmennettua tietokannan toteutustyökaluksi MS Accessin, mistä syystä oli luontevaa lähteä suunnittelemaan myös tulostuvaa asiakasraporttia Accessin omilla työkaluilla.

Kuvassa 6 osoitetaan numerolla 1 lisätyt raportit. Numerolla 2 kuvaan on merkitty Accessin tarjoamia työkaluja. Kuten lomakkeille, on Accessiin sisäänrakennettu useita erilaisia sisältö- ja asemointielementtejä myös raporttien käsittelyyn. Kuvaesimerkissä on muokattavaksi avattuna numerolla 3 merkitty pääraportti, johon on sisällytetty alaraportteja. Numerolla 4 kuvaan on merkitty raportteihin lisätyille objekteille löytyvät ulkoasu- ja toiminta-asetukset.



Kuva 6. Accessin työkalut raportin rakentamiseen.

Projektin raportti siis rakennettiin pääraportista ja tähän pääraporttiin sisällytetyistä alaraporteista. Pääraporttiin määriteltiin koko tulostuvaa raporttia koskevat ulkoasu- ja toimintaominaisuudet, alaraportit sisälsivät kukin aina

pienen, yksittäisen osan raporttia. Kokonaisuudessaan raportti koostuu siis pääraportista "RAPORTTI", sekä alaraporteista "subrArvioitsija", "subrAsiakas", "subrKohdeyht", "subrKuvat", "subrTarkastusraportti" ja "subrYhteenveto". Alaraporttiin "subrArvioitsija" tulostuvat siis esimerkkeinä käyttöliittymäpuolella "Raportti"-lomakkeeseen valitut raportin tekijän tiedot, alaraporttiin "subrAsiakas" lomakkeeseen "Asiakas" kirjatut tiedot. Olen sisällyttänyt ohjelmiston viimeisimmästä versiosta 0.9.0:sta tulostetun malliraportin tämän opinnäytetyön liitteeksi, jotta raporttikokonaisuus olisi ulkopuolisen helpompi hahmottaa (Liite 3: Malliraportti).

Tarkastusraportti					
	Kunto	Lisätutkimus	Toimenpide	Aikataulu	Määrä Yks.
KATON YLEISKUNTO					
Ullakko/tuuletustila	Tyydyttävä	Ei	Erityisseuranta	0-3v.	
<i>Vinkuhormin painetappi reistaa</i>					
Tuuletus	Valttava	Kyllä	Korjataan	0-3v.	
Katteen alusta	Hyvä	Ei	Erityisseuranta	0-3v.	
Aluskate	Tyydyttävä	Ei	Erityisseuranta	0-3v.	3 m²
Kate	Hyvä	Ei	OK		
Katteen kiinnitys	Hyvä	Ei	OK		4 m²
Jiirit	Hyvä	Ei	OK		2 m²
Kaltevuudet	Tyydyttävä	Ei	Erityisseuranta	0-3v.	2 m²
Räystäät	Hyvä	Ei	OK		4 jm
KÄYTTÖ- JA HUOLTOTURVALLISUUS					
Talotikkaat	Tyydyttävä	Ei	Erityisseuranta	0-3v.	2 kpl
Lumiesteet	Valttava	Ei	Korjataan	Heti	2 m²
<i>Kesäaikoina turhat</i>					
Kulkusillat	Huono	Ei	Uusitaan	Heti	2 m²
Turvakiskot, kiskovaunut	Tyydyttävä	Ei	Erityisseuranta	0-3v.	3 kpl
Lapetikkaat	Tyydyttävä	Ei	Erityisseuranta	0-3v.	

Kuva 7. Tarkastusraportin muotoilu versioissa 0.1.0 - 0.6.0.

Kehitystyön suurin painoarvo ensimmäisten kehitysversioiden aikana oli tietokannan ja raportin suunnittelulla, sai raportti jo varhaisessa vaiheessa projektin kehityskaarta lähes lopullisen muotonsa. Suurimmat muutokset versiosta olivatkin jo valtaosin ulkoasullisten yksityiskohtien työstämistä. Esimerkiksi kuvassa 7 esitellyn tarkastusraportti-osion värikoodaus ei ollut

prioriteetiltaan tärkeä, joten se jätettiin projektin loppupäähän. Malliesimerkki osion lopullisesta ulkoasusta on löydettävissä liitteeksi lisätystä malliraportista (Liite 3:Malliraportti).

</

Kuva 8. Kustannusarvio versiossa 0.6.0.

Varhaisimpiin raportin versioihin ei toki vielä kaikkia osuuksia saatu toimivina mukaan. Merkittävimpänä esimerkkinä kuvassa 8 esitetty raportin yhteenvetosivulle tulostuva kustannusarvio.

4 YHTEENVETO JA POHDINTA

Opinnäytetyön tavoitteena oli määritellä ja suunnitella toimeksiantajan raportointikäytäntöjä virtaviivaistava ja yhdenmukaistava työkalu aiemmin käytettyjen sekalaisten työkalujen ja -menetelmien korvaajaksi. Työkalulle määriteltiin vaadittavat ominaisuudet yhteistyössä toimeksiantajan edustajan kanssa. Tässä vaiheessa apuna käytettiin tarkentavien keskustelujen ja aiempien raporttien analysoinnin lisäksi myös tulevan työkalun toimintaa havainnollistavaa prototyyppiä, joka auttoi selvittämään useita erityistä huomiota vaativia ominaisuuksia. Toimeksiantajan palaute oli prototyypin ja ensimmäisten kehitysversioiden perusteella positiivista. Mikäli määritellyt ominaisuudet saataisiin toteutettua suunnitelman mukaisesti, tulisi työkalu varmasti käyttöön ja säästäisi huomattavan määrän työtunteja jatkossa.

4.1 Projektin haasteet

Ohjelmiston määrittelyvaihe sujui varsin kivuttomasti ja nopeasti. Tuottamani käyttöliittymän prototyyppi auttoi hahmottamaan toimeksiantajan todellisia tarpeita ja helpotti kommunikointia. Prototyypistä saatiin vahva pohja eri näkymistä ja ominaisuuksista myös lopulliseen käyttöliittymään, vaikka muutoksiakin toki jouduttiin tekemään. Lisäksi tietokannan rakenteeseen saatiin paljon hyvää pohjatietoa toimeksiantajan aiemmin tuottamien raporttien perusteella. Tätä samaa pohjatietoa voitiin käyttää myös työkalusta tuotettavien uusien raporttien suunnitteluun. Monia aiempien raporttien osioita saatiin siirrettyä tulevaan raporttipohjaan käytännössä suoraan, joitakin puolestaan kehitettiin eteenpäin luettavamman ja ammattimaisemman vaikutelman luomiseksi.

Projektiä hidastaneet ongelmat astuivat vastaan määrittely- ja suunnittelupöydältä ohjelmointitoteutukseen siirryttäessä. Jälkikäteen arvioituna projekti olisi ollut ehkä järkevämpi toteuttaa kahden työntekijän voimin

suurempien työtuntien mahdollistamiseksi ja ohjelmoidessa vastaan tulleiden hidasteiden minimoimiseksi.

Tietokannan, käyttöliittymän ja raporttien suunnittelusta Access-ympäristössä minulla oli jo entuudestaan jonkin verran kokemusta, mutta ohjelmointiosaamiseni ja kehitystyökalulle ominaisten, aiemmasta osaamisesta poikkeavien erikoistoimintojen käyttö vaativat runsaasti sekä uuden opettelua, että vanhan kertaamista.

Vaikka teoriaa lukiessa monet ohjelmistotuotannon protoilumallia käsittelevät aineistot varoittelivatkin eräänlaisesta todellisuuden hämartymisestä, tämä kuitenkin ehkä unohtui tosielämässä. Sekä määrittelyssä käytetty poisheitetty prototyyppi, että tuotantoympäristöön rakennettavan evolutiivisen prototyypin sinänsä toimivan näköiset ruudut ja navigaatio pääsivät luomaan valheellista valmiuden tunnetta, vaikka ohjelmistosta puuttuikin vielä valtaosa kaikesta ohjelmoinnillisesta toiminnallisuudesta.

Tuotantovaihetta toteutettiin pääasiallisesti etätyöskentelynä, mikä asetti omat haasteensa. Kommunikointi oli molemminpuolisesti katkonaista, mikä luonnollisesti aiheutti epätietoisuutta kummallekin taholle. Suurimmaksi haasteeksi muodostuivat lopulta kuitenkin ehkä työskentelyn osa-aikaisuus ja yksityiselämässä tapahtuneet muutokset. Projektin ulkopuolinen palkkatyö jouduttiin priorisoimaan opinnäytetyön yläpuolelle välillä pitkiksikin ajanjaksoiksi, mikä valitettavasti aiheutti suuria venymiä aikatauluihin. Nämä venymät hankaloittivat myös aktiivista keskustelua toimeksiantajan kanssa. Aikatauluongelmat luonnollisesti laskivat sekä omaa, että toimeksiantajan luottoa projektin valmistumiseen ajallaan.

Projektia jälkikäteen arvioidessani löydänkin asioita, joita olisi pitänyt tehdä toisin. Määrittely- ja suunnitteluvaiheissa aikataulutus ja projektinhallinta kokonaisuudessaan toimivat juuri niin sulavasti kuin niiden pitikin. Muutoksiin voitiin reagoida nopeasti ja dialogi toimeksiantajan ja itseni välillä oli aktiivista. Ohjelmointivaiheeseen päästessä tämä töiden tehokas rytmitys kuitenkin pääsi häviämään. Yhtenä osasyynä tähän on varmasti oman ohjelmointiymmärrykseni

rajallisuus ja osin uusi toteutusympäristö. Vaikka Access tietokantatyökaluna minulle olikin jo entuudestaan tuttu, en ollut sillä aiemmin toteuttanut käyttöliittymiä, ohjelmointia, enkä raportteja. Läpi projektin on siis samaan aikaan sekä opiskeltu, että toteutettu kokonaisuutta. Virheet ja niiden korjaamiseen mennyt aika veivät täten turhan suuren osan niistä työtunneista, joita projektiin käytettiin.

4.2 Työn lopputulos ja jatkokehityksen tarve

Ohjelmiston viimeisin versio on toimitettu asiakkaalle tammikuussa 2014. Tässä versiossa oltiin saatu toimintaan osa aiemmasta joulukuisesta versiosta vielä puuttuneita ominaisuuksia ja korjattua työkalun toimintaan vaikuttaneita ohjelmointivirheitä. Kaikkia projektin alussa määriteltäviä tavoitteita tässä viimeisimmässä versiossa ei vielä saatu toteutettua, esimerkiksi jo valmiiden raporttien kannasta etsimiseen käytettävien hakuehtojen määrä on vähäisempi kuin suunniteltua. Käyttöliittymä ei vielä myöskään osaa halutulla tavalla selventää käyttäjälle niitä raportin osioita, joissa vielä on täydennettävää.

Suurimmiksi poikkeamiksi alkuperäisen suunnitelman ja vaatimusmäärittelyn pohjalta jäivät kuitenkin tarkastusraporttiosion oletusarvoisten kategorioiden ja alakohtien toteutus, sekä käyttäjäoikeuksien toteutus. Tietokannan ylläpitoa ja vakioarvojen muokkaamista ei vielä saatu rajattua vain määrätyille käyttäjille. Suunnitelman mukaan kullekin raportille oltaisi voitu valita vain tarpeelliset kategoriat ja niihin tarpeelliset alakohdat. Tammikuun versiossa tätä ei vielä saatu täysin halutulla lailla toimimaan, vaan kaikkiin raportteihin tulostuvat kaikki kannasta löytyvät kategoriat alakohtineen, vaikka niille ei kyseisessä raportissa mitään sisältöä olisi merkittykään. Vaikka kyseessä on pohjimmiltaan enemmän visuaalinen epäkohta, olisi suunniteltu toteutustapa varmasti tarjonnut raportin saaville asiakkaille paljon selkeämpää luettavaa ja täten paremman asiakaskokemuksen.

Näiden kehityskohteiden lisäksi ohjelmistokoodiin olisi hyvä saada vielä laajemmin arvojen syöttöjä tarkistavia ja täten kannan sisältämän tiedon

ehjyyttä ja käyttökelpoisuutta parantavia ominaisuuksia. Säännöllisiä, ohjelmiston toimintaa rikkovia virheitä ei viimeisimmässä versiossa oman testaamiseni aikana enää ilmennyt.

Tästä viimeisimmästä, tammikuun versiosta en koskaan saanut palautetta, tai edes vahvistusta yhteyshenkilöltäni, mistä syystä en tiedä onko ohjelmisto kenties otettu yrityksessä käyttöön, vai onko koko projekti viivästyneenä hylätty.

4.3 Yhteenveto

Projekti opetti minulle paljon uutta ohjelmistokehityksestä, niin teorian, kuin käytännön toteutuksenkin puolelta. Olen saanut laajentaa teknistä osaamistani entuudestaan tuttujen työkalujen minulle uusiin ominaisuuksiin, samoin kuin olen voinut vahvistaa itselleni mieleisten määrittely- ja suunnittelupuolten osaamista suuresti. Koin prototyyppimallin hyödyntämisen tuoneen ohjelmiston vaatimusmäärittelyyn selkeyttä ja ytimekkyyttä ja olenkin ylpeä että ajatus tämän työmallin hyödyntämisestä heräsi jo ennen kuin olin siitä oikeasti lähtenyt teoriatasolla lukemaan.

LÄHTEET

Haikala, I. & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Talentum Media Oy.

Microsoft 2014. Tietokannan normalisoinnin perusteiden kuvaus. Viitattu 12.6.2014.
<http://support.microsoft.com/kb/283878/fi>.

Niemi, O. 2013. Tietokantasovelluksen suunnittelu ja toteutus RP-Elementti Oy:lle. Viitattu 11.6.2014. <http://urn.fi/URN:NBN:fi:amk-2013061213928>.

Paakki, J. 2011. Ohjelmistojen vaatimusmäärittely. Helsingin yliopisto, tietojenkäsittelyn laitos. Viitattu 7.6.2014. <http://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf>.

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Docendo Finland Oy.

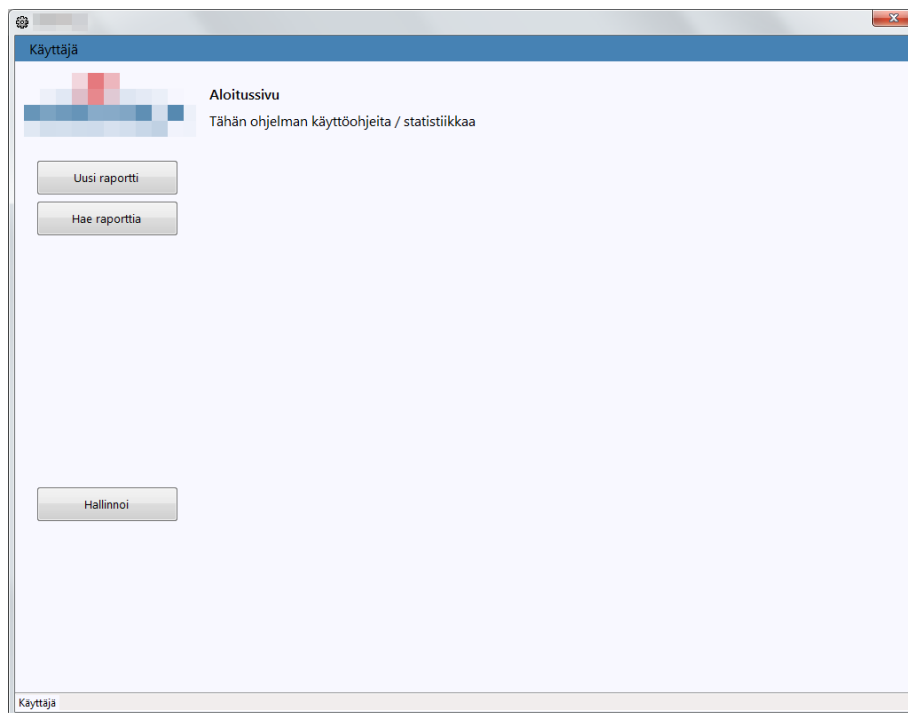
Wikipedia 2014a. Software prototyping. Viitattu 13.3.2014, 27.4.2014, 3.6.2014.
http://en.wikipedia.org/wiki/Software_prototyping.

Wikipedia 2014b. Software requirements specification. Viitattu 5.6.2014.
http://en.wikipedia.org/wiki/Software_requirements_specification.

Wikipedia 2014c. Functional requirement. Viitattu 5.6.2014.
http://en.wikipedia.org/wiki/Functional_requirements.

Wikipedia 2014d. Non-functional requirement. Viitattu 5.6.2014, 7.6.2014.
http://en.wikipedia.org/wiki/Non-Functional_Requirements.

Prototyypin esittely asiakkaalle



Tarkoituksena on ollut pitää käyttöliittymä mahdollisimman yksinkertaisena ja helppokäyttöisenä. Ruudulla on kerrallaan näkyvissä vain ne ominaisuudet/työkalut, joita kyseisessä työvaiheessa tarvitaan.

Ohjelman etusivulta pääsee lisäämään uutta raporttia, hakemaan aiempaa tulostusta tai muokkausta varten, sekä pääkäyttäjän hallintopaneeliin (joka ei tässä käyttöliittymä-demossa vielä ole mukana).

Tarkastusraportti-välilehdelle siirryttäessä sivuvalikkoon aukeaa uusia vaihtoehtoja, uuden kategorian lisääminen ja valitun kategorian tietojen muokkaaminen.

Lisää uusi aihe antaa käyttäjän lisätä tarkastusraporttiin uusia ominaisuuksia (tätä kirjoittaessa huomaa ja muistan että aihealueiden poisto-ominaisuus uupuu demosta). Tallennettaessa luotu uusi ominaisuus päivittyisi listaan ja sen ominaisuuksia ja arvoja pääsisi muokkaamaan **Muokkaa valittua**-napilla.

Käyttäjä

Hakurajaus: Kaikki Hakuarvo: Hae

Pvm	Raportin tekijä	Kohteen nimi	Osoite
-----	-----------------	--------------	--------

Tallenna

Muokkaa valittua

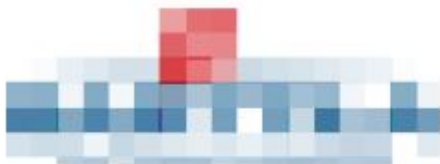
Tulosta

Sulje

Käyttäjä

Hae raporttia näyttää tässä demoversiossa vääränlaisen sivuvalikon. Oikeassa versiossa tuo "Tallenna" jää pois ja Muokkaa valittua ohjaa oikeaan kohtaan (avaa valitun raportin tiedot samanlaisessa näkymässä, kuin mitä uuden raportin lisäys-sivu on.

Malliraportti



Raportti 1

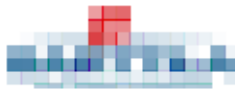


Testikohde 1

YRITYS OY
Osollteenkatu 2 A
24280 SALO

internet:
etu.sukunimi@yritys.fi
www.yritys.fi

puh. 012 345 6789
fax. 012 345 6789
gsm 012 345 6789



Raportti 1

Kuntoarvioitsija

Yhteyshenkilö: Testikäyttäjä 4
 Puhelin: TK_numero4
 Sähköposti: TK_maili4
 Arvio tehty: 22.8.2013

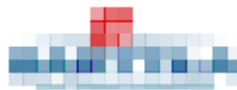
Asiakas

Yritys: Testiasiakas 1
 Yhteyshenkilö: Testiyhteyshenkilö 1
 Yhteyshenkilö: TYH_puh 1
 Yhteyshenkilö: TYH_maili 1
 Lisätietoja:
 Testiasiakas 1:n lisätiedot

Yleiskuvaus kohteesta

Kohteen nimi: Testikohde 1
 KohdeOsoite: Testiosoite 1
 Postinumero: 24280 Postitoimipaikka: SALO
 Kuvaus aiemmista vuodoista / vaurioista
 Ei aiempia

Erilliset kattotasot (kpl)	Asuinrakennukset (kpl)	Huoltorakennukset (kpl)
2	1	1



Raportti 1

Yhteenveto

Alkuteksti raportille 1

Kuva 1

Raportti 1 Kuva 1 Kuvateksti 2

Kuva 2

Raportti 1 Kuva 2 Kuvateksti 2

Kuva 3

Raportti 1 Kuva 3 Kuvateksti 2

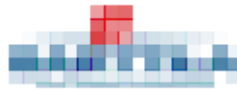
Kuva 4

Raportti 1 Kuva 4 Kuvateksti 2

Lopputeksti raportille 1

Kustannusarvio

Välitön	Selite 1	200 €
Suosittelava	Selite 2	150 €
Välitön	Selite 3	65 €
Välitön	Selite 4 muok s	22000 €
Välitön	Selite 7	2500 €
Välitön	Selite 16	314 €
Välitön	Selitetesti	500 €



Raportti 1

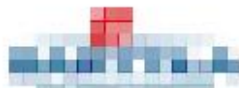
Tarkastusraportti

	Kunto	Lisätutkimus	Toimenpide	Alkataulu	Määrä Yks.
KATON YLEISKUNTO					
Ullakko/tuuletustila	Tyydyttävä	Ei	Erityisseuranta	0-3v.	
<i>Vinku hormin painetappi reistaa</i>					
Tuuletus	Välttävä	Kyllä	Korjataan	0-3v.	
Katteen alusta	Hyvä	Ei	Erityisseuranta	0-3v.	
Aluskate	Tyydyttävä	Ei	Erityisseuranta	0-3v.	3 m²
Kate	Hyvä	Ei	OK	>5 v.	
Katteen kiinnitys	Hyvä	Ei	OK		4 m²
Jiirit	Hyvä	Ei	OK		2 m²
Kaltevuudet	Tyydyttävä	Ei	Erityisseuranta	0-3v.	2 m²
Räystäät	Hyvä	Ei	OK		4 jm
KÄYTTÖ- JA HUOLTO TURVALLISUUS					
Talotikkaat	Tyydyttävä	Ei	Erityisseuranta	0-3v.	2 kpl
Lumiesteet	Välttävä	Ei	Korjataan	Heti	2 m²
<i>Kesäaikoina turhat</i>					
Kulkusillat	Huono	Ei	Uusitaan	Heti	2 m²
Turvakiskot, kiskovaunut	Tyydyttävä	Ei	Erityisseuranta	0-3v.	3 kpl
Lapetikkaat	Tyydyttävä	Ei	Erityisseuranta	0-3v.	
Kattopollarit	Tyydyttävä	Ei	Erityisseuranta	0-3v.	
<i>Pitävät rosvoit loitolla</i>					
LÄPIMENNIT					



Raportti 1

Savupiiput	Hyvä	Ei	OK		
IV-hormit	Välttävä	Kyllä	Korjataan	0-3v.	5 kpl
IV-tarkastusluukut	Tyydyttävä	Ei	Erityisseuranta	3-5v.	4 m²
Kattoluukut	Hyvä	Ei	OK		3 m²
Kattoikkunat	Huono	Ei	Uusitaan	Heti	5 m²
Putket	Välttävä	Ei	Korjataan	0-3v.	4 jm
Antennit	Tyydyttävä	Ei	Erityisseuranta	3-5v.	
Alipainetuulettimet	Tyydyttävä	Ei	Erityisseuranta	3-5v.	2 m²
VEDENPOISTOJÄRJESTELMÄ					
Sadevesikaivot	Tyydyttävä	Ei	Erityisseuranta	3-5v.	
Ulosheittäjät	Hyvä	Ei	OK		2 m²
Vesikourut	Välttävä	Kyllä	Korjataan	0-3v.	2 m²
Syöksytorvet	Hyvä	Ei	OK		4 m²
PELLITYKSET					
Räystäät	Tyydyttävä	Ei	Erityisseuranta	3-5v.	5 m²
Lipat	Tyydyttävä	Ei	Erityisseuranta	>5 v.	
Läpiviennit	Tyydyttävä	Ei	Erityisseuranta	3-5v.	



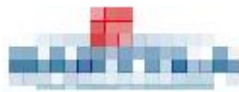
Raportti 1

KATON YLEISKUNTO

ullakko/tuuletustila, tuuletus, katteen alusta, aluskate, kate, jirit, ruuvit, ylösnostot, kaltevuus, räystäät



Raportti 1 Kuva 1 Kuvateksti 1



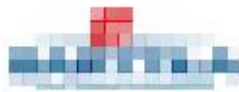
Raportti 1

KÄYTTÖ- JA HUOLTOTURVALLISUUS

talotikkaat, lumiesteet, kulkusillat ja turvakiskot, lapetikkaat, kattopollarit



Raportti 1 Kuva 2 Kuvateksti 1



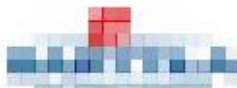
Raportti 1

LÄPIMENNIT

savupiiput, IV-hormit, IV-tarkistusluukut, kattoluukut, kattoikkunat, putket, antennit, alipainetuulettimet



Raportti 1 Kuva 3 Kuvateksti 1



Raportti 1

KATON YLEISKUNTO

ullakko/tuuletustila, tuuletus, katteen alusta, aluskate, kate, jiirit, ruuvit, ylösnostot, kaltevuus, räystäät



Raportti 1 Kuva 4 Kuvateksti 1